Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server

Wu-chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole

Goal

- Understand the resource requirements of a popular on-line FPS (first-person shooter) game

Why games?

- Rapidly increasing in popularity
  - Forrester Research: 18 million on-line in 2001
  - Consoles on-line
    - Playstation 2 on-line (9/2002)
    - Xbox Live (12/2002)
  - Cell phones
    - Nokia Doom port (yesterday)

Why FPS?

- Gaming traffic dominated by first-person shooter genre (FPS) [McCreary00]
**Networked FPS lineage**

**Doom**
- **Doom II**
  - Quake
  - + Quake/World variants
  - + Team Fortress
  - + Capture the Flag
  - + Quake II
  - + Soldier of Fortune
  - + Heretic II
  - Quake III Arena
    - + Medal of Honor: Allied Assault
    - + Return to Castle Wolfenstein
    - + Soldier of Fortune 2
    - + Jedi Knight II
  - Doom III

**Unreal**
- **Unreal Tournament**
  - Unreal Tournament 2003
    - + America's Army: Operations

**Half-Life**
- **Counter-Strike**
  - + Counter-Strike
  - + Day of Defeat
  - + Urban Terror
  - + Team Fortress Classic
  - + Team Fortress 2

8 of top 10 games derived from one of two lineages
About the game...

- Half-Life modification
- Two squads of players competing in rounds lasting several minutes
- Rounds played on maps that are rotated over time
- Each server supports up to 32 players

Centralized server implementation
- Clients update server with actions from players
- Server maintains global information and determines game state
- Server broadcasts results to each client

Sources of network traffic
- Real-time action and coordinate information
- Broadcast in-game text messaging
- Broadcast in-game voice messaging
- Customized spray images from players
- Customized sounds and entire maps from server

The trace

- cs.mshmro.com (129.95.50.147)
  - Dedicated 1.8GHz Pentium 4 Linux server
  - OC-3
  - 70,000+ unique players (WonIDs) over last 4 months
- One week in duration 4/11 – 4/18
- 500 million packets
- 16,000+ sessions from 5800+ different players

A week in the life...
Variance time plot

Normalized to base interval of 10ms

Digging deeper

- Periodic server bursts every 50ms
  - Game must support high interactivity
  - Game logic requires predictable updates to perform lag compensation

Finding the source of predictability

- Games must be fair across all mediums (i.e. 56kers)
  - Aggregate predictability due to “saturation of the narrowest last-mile link”
- Histogram of average per-session client bandwidth

Digging deeper

- Low utilization every 30 minutes
  - Server configured to change maps every 30 minutes
  - Traffic pegged otherwise....
**Packet sizes**

- Supporting narrow last-mile links with a high degree of interactivity requires small packets
  - Clients send small single updates
  - Servers aggregate and broadcast larger global updates

**Implications**

- Routers, firewalls, etc. must be designed to handle large bursts at millisecond levels
  - Game requirements do not allow for loss or delay (lag)
  - Should not be provisioned assuming a large average packet size [Partridge98]
  - If there are buffers anywhere, they must...
    - Use ECN
    - Be short (i.e. not have a bandwidth-delay product of buffering)
    - Employ an AQM that works with short queues

**Implications**

- ISPs, game services
  - Must examine “lookup” utilization in addition to link utilization
  - Concentrated deployments of game servers may be problematic
    - Large server farms in a single co-lo
    - America's Army, UT2K3, Xbox

**On-going work**

- Other pieces in the provisioning puzzle
  - Aggregate player populations
  - Geographic distributions of players over time (IP2Geo)
- Impact on route and packet classification caching
- Other FPS games
  - HL-based: Day of Defeat
  - UT-based: Unreal Tournament 2003, America's Army
  - Quake-based: Medal of Honor: Allied Assault
  - Results apply across other FPS games and corroborated by other studies
Future work

- Games as passive measurement infrastructure
  - Only widespread application with continuous in-band ping information being delivered (measurement for free)
  - “Ping times” of all clients broadcast to all other clients every 2-3 seconds
  - 20,000+ servers, millions of clients
- Games as active measurement infrastructure
  - Thriving FPS mod community and tools
  - Server modifications [Armitage01]
Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server

Wu-chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole

OGI SCHOOL OF SCIENCE & ENGINEERING
OREGON HEALTH & SCIENCE UNIVERSITY
Goal

• Understand the resource requirements of a popular on-line FPS (first-person shooter) game
Why games?

• Rapidly increasing in popularity
  – Forrester Research: 18 million on-line in 2001
  – Consoles on-line
    • Playstation 2 on-line (9/2002)
    • Xbox Live (12/2002)
  – Cell phones
    • Nokia Doom port (yesterday)
Why FPS?

- Gaming traffic dominated by first-person shooter genre (FPS) [McCreary00]
Why CS?

Serverspy FPS rankings (10/31/2002)

- Half-Life
- MedalOfHonor: Allied Assault
- Quake III Arena
- Battlefield 1942
- Unreal Tournament
- Return to Castle Wolfenstein
- Unreal Tournament 2003
- Soldier of Fortune 2: Double Helix
- America’s Army: Operations
- Neverwinter Nights

# of players
Why CS?

Serverspy HL mod rankings (10/31/2002)

- Counter-Strike
- Day of Defeat
- Team Fortress Classic
- Deathmatch
- The Specialists
- Firearms
- SvenCo-op
- Vampire Slayer
- Front Line Force
- Action Half-Life

# of players
Networked FPS lineage

Doom

Doom II

Quake
  + QuakeWorld variants
  + Team Fortress
  + Capture the Flag

Quake II
  + Soldier of Fortune
  + Heretic II

Quake III Arena
  + Medal of Honor Allied Assault
  + Return to Castle Wolfenstein
  + Soldier of Fortune 2
  + Jedi Knight II

Doom III

Unreal

Unreal Tournament

Unreal Tournament 2003
  + America's Army: Operations

8 of top 10 games derived from one of two lineages
Counter-Strike
About the game...

• Half-Life modification
• Two squads of players competing in rounds lasting several minutes
• Rounds played on maps that are rotated over time
• Each server supports up to 32 players
About the game...

• Centralized server implementation
  – Clients update server with actions from players
  – Server maintains global information and determines game state
  – Server broadcasts results to each client

• Sources of network traffic
  – Real-time action and coordinate information
  – Broadcast in-game text messaging
  – Broadcast in-game voice messaging
  – Customized spray images from players
  – Customized sounds and entire maps from server
The trace

- cs.mshmro.com (129.95.50.147)
  - Dedicated 1.8GHz Pentium 4 Linux server
  - OC-3
  - 70,000+ unique players (WonIDs) over last 4 months
- One week in duration 4/11 – 4/18
- 500 million packets
- 16,000+ sessions from 5800+ different players
A week in the life...
Variance time plot

Normalized to base interval of 10ms
Digging deeper

- Periodic server bursts every 50ms
  - Game must support high interactivity
  - Game logic requires predictable updates to perform lag compensation
Digging deeper

- Low utilization every 30 minutes
  - Server configured to change maps every 30 minutes
  - Traffic pegged otherwise....

---

Interval size=1sec  
Interval size=30min
Finding the source of predictability

- Games must be fair across all mediums (i.e. 56kers)
  - Aggregate predictability due to “saturation of the narrowest last-mile link”
- Histogram of average per-session client bandwidth
Packet sizes

- Supporting narrow last-mile links with a high degree of interactivity *requires* small packets
  - Clients send small single updates
  - Servers aggregate and broadcast larger global updates
Implications

- Routers, firewalls, etc. must be designed to handle large bursts at millisecond levels
  - Game requirements do not allow for loss or delay (lag)
  - Should not be provisioned assuming a large average packet size [Partridge98]
  - If there are buffers anywhere, they must...
    - Use ECN
    - Be short (i.e. not have a bandwidth-delay product of buffering)
    - Employ an AQM that works with short queues
Implications

• ISPs, game services
  – Must examine “lookup” utilization in addition to link utilization
  – Concentrated deployments of game servers may be problematic
    • Large server farms in a single co-lo
    • America's Army, UT2K3, Xbox
On-going work

• Other pieces in the provisioning puzzle
  – Aggregate player populations
  – Geographic distributions of players over time (IP2Geo)

• Impact on route and packet classification caching

• Other FPS games
  – HL-based: Day of Defeat
  – UT-based: Unreal Tournament 2003, America's Army
  – Quake-based: Medal of Honor: Allied Assault
  – Results apply across other FPS games and corroborated by other studies
Future work

- Games as passive measurement infrastructure
  - Only widespread application with continuous in-band ping information being delivered (measurement for free)
  - “Ping times” of all clients broadcast to all other clients every 2-3 seconds
  - 20,000+ servers, millions of clients

- Games as active measurement infrastructure
  - Thriving FPS mod community and tools
  - Server modifications [Armitage01]
Questions?